Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0008

Controlling the operation of the dc motor by using pid with metaheuristic technology

Khalaf Abdullah Khalaf KHALAF Mustafa Teke

Department of Electrical and Electronics Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkey

Cite this article: Khalaf KAK, Teke M. Controlling the operation of the dc motor by using pid with metaheuristic technology. *J Comp Electr Electron Eng Sci*. 2023;1(2):34-40.

Corresponding Author: Khalaf Abdullah Khalaf KHALAF, khalafabd1994@gmail.com

Received: 29/07/2023 ◆ **Accepted**: 04/10/2023 ◆ **Published**: 31/10/2023

ABSTRACT

This paper presents an investigation into precise trajectory tracking and synchronization of two-axes direct current (DC) motor control, with an emphasis on a cascade proportional-plus-integral (P-PI) controller to regulate the speed and position of a single-axis permanent magnet DC (PMDC) motor. Various methods were explored for the controller's design process, including classical methods (CM) and three optimization strategies: genetic algorithm (GA), dandelion optimization algorithm (DOA), and butterfly optimization algorithm (BOA), with the latter found to be the most effective. Simulation was a crucial component in assessing the efficiency of these methods. A comparative analysis of four tuning strategies (CM, GA, BOA, and DOA) was conducted to ascertain optimal settings for the P-PI cascade controller. The DOA outperformed the others, providing accurate tracking with no deviation from the reference location or speed overshoot. Moreover, DOA ensured safety by limiting voltage and current to prevent potential damage to the motor. The findings thus suggest that the proposed P-PI controller with DOA can serve as a reliable solution for speed and position control in single-axis PMDC motors.

Keywords: PMDC, fine-tuning the PI controller, BOA, DOA, PID controllers.

INTRODUCTION

In its most basic form, an electric motor is a mechanism that uses electricity to create motion (Bigelow, 2020; Franchi and Claiton, 2019). This generated motion can be exploited in various ways for a variety of applications. This study was done by using one of the types of DC motors, which is PMDC. This motor is used in many applications and is similar in structure and operation to the shunt-connected DC motor (Momoh and James, 2018; Krause et al., 2002). On the other hand, to control the most important functions of PMDC, namely velocity and position, a cascade control system consisting of three controllers were used to control the current, speed, and position (Son and Young 2014; Cankurtaran et al., 2019). This controller provides the required response to the system, minimizes error, and returns to a steady state when a specified load is applied. The main reason for using such controllers and other controllers is to use them to control many motor functions such as speed, current, and position that humans cannot control manually (Wang and Liuping, 2020; Raja et al., 2017). In reference (Cuong, 2013), the model was developed by connecting two (dual) motors instead of one DC motor to take advantage of them in many applications, such as sharing a specific load. This model can also be used in many applications, some of which require precise speed control, such as paper and textile mills, and others require high and accurate position control, such as metal cutting machines, CNC machines, and robotics (Tang, 2001; Romero et al., 2006).This work has been implemented in a real-time Matlab environment, this study demonstrates a remarkable degree of precision in guiding the motor to its target location at a constant speed, irrespective of the presence or absence of a load. Simulation results obtained from Matlab verified that the PMDC motor could be controlled with exceptional accuracy, fulfilling the desired position, while maintaining an accurate tracking trajectory. A comprehensive review of related literature confirms that a variety of controller types, including PID, fuzzy logic, and 2-SMC controllers, can manage the PMDC motor effectively. In this study, a cascade control system, characterized by its flexibility and robustness, is adopted for this study. The cascade controller employed in this investigation comprises three individual controllers: the position, speed, and current controllers.

Literature Review

Tang (2001) proposed a PID controller architecture for managing the speed and position of a DC motor, which employs a low-cost digital signal processor, the TMS320C31 suite. Real-time values of speed and position controller parameters were set in a direct online manner during the operation of the DC motor to ensure its continuous operation without shutdown. Romero and Concha (2006) presented a system for controlling the velocity and position of a moving robot using a brushless DC motor, with a motor

This work is licensed under a Creative Commons Attribution 4.0 International License.

connected to each wheel. A three-phase bridge was designed using N- Mosfet, along with an electronic circuit to drive the motors. This paper references several studies on motor control methodologies. Tang (2001) implemented a PID controller architecture utilizing a low-cost digital signal processor, the TMS320C31 suite, to manage the DC motor's speed and position. Outputs from Hall sensor and optical encoder were also harnessed in conjunction with pulse width modulation to form closed-loop control for velocity and position. Pisano et al. (2008) developed a cascade control system for managing the speed and location of the permanent magnet DC motor. Despite uncertainties in motor and load parameters, the application of a second-order sliding mode controller (2-SMC) ensured excellent performance and precise tracking. Talavera et al. (2014) presented a bidirectional DC motor speed and position control system constructed using ATMEGA32 microcontrollers and LabVIEW. An optical encoder attached to the motor shaft collected information on rotational speed and angular position, and the motor was powered using the PWM technique. A graphical user interface in LabVIEW software was used to input the reference signal, be it a velocity signal or a position angle. Taha et al. (2015) reported a cascade PI(D) control system for the speed and position of the permanent magnet DC motor. PID gains were adjusted using three distinct methods: a traditional manner and two ideal ones. These strategies were then compared to identify the one providing the most reliable velocity and position regulation. Taut and Marius (2018) introduced a closed-loop DC motor speed and position control system comprising a DC actuator motor, a position sensor, and the mechanical load to be moved by the motor. This system's control methodology was realized using Simulink and embedded code. Syh-Shiuh and Pau (2000) proposed an integrated movement system to enhance the performance of a CNC machine in terms of path tracking, which included feedforward and feedback with multi-axis cross-coupled control (CCC). This controller was efficiently designed through a comprehensive system analysis and a novel formulation of the contouring error transfer function (CETF) for multi-axis systems. Ishizaki et al. (2014) implemented a speed and position control system for two PMDC motors used in a gantry-style machine tool. Each PMDC motor in this setup was managed by two separate controllers, a position (P) controller and a speed (PI) controller. A cross-coupling approach was employed to interconnect the motors, and a methodology was created to determine the values of the system parameters for precise synchronization of motor movement along the axes. Panlong and Wang (2019) proposed an autonomous guided vehicle powered by two separate DC motors. Two control methods were used: parallel PID control and coupled PID control of deviation. These methods were evaluated using MATLAB simulations and KINECT sensor experiments. The coupled PID controller of deviation performed marginally better than the parallel PID controller in a transport scenario. Lastly, (Ali et al., 2019) demonstrated that two DC motors could be controlled in parallel using a Hall sensor. Precise control of the motors' speeds and positions was necessary to achieve the desired parallel positioning. Position and speed were estimated using the back EMF compensator and the current model speed observer, respectively. Position discrepancies between the motors could be adjusted using the instantaneous position compensator, and the speed could be approximated using the average speed between the speed observer and the actual speed.

METHODS

The PMDC (Permanent Magnet Direct Current) motor is a significant component in many applications due to its simple construction and reliable performance. Understanding its construction and the principles behind its operation is essential for those involved in its application and further development. A typical PMDC motor consists of two main parts: the stator and the rotor (also known as the armature). The stator forms the stationary part of the motor, providing the magnetic field. It's a steel cylinder with permanent magnets attached to its inner surface. These magnets are oriented so that their poles are facing inward toward the rotor. The magnetic poles of the stator are fixed in such a way that they face the armature, providing the necessary magnetic field for operation. The rotor or armature, the moving part of the motor, is composed of a winding of wire placed around an iron core. This armature winding is connected to a commutator, which is a device that switches the direction of current flow through the windings as the rotor turns. This switching action ensures that the rotor continues to turn in the same direction. The commutator brushes, usually made of carbon, maintain the electrical contact with the rotating commutator.

PMDC Mathematical Model

The circuit is powered by a voltage source linked to the armature coil. The electrical elements of this circuit encompass the inductance (La) and armature resistance (Ra), wired in series. Additionally, a counter-electromotive force or back emf (Ea), which emerges once the coil spins and intersects the flux lines produced by the permanent magnet, opposes the voltage source's direction. The mechanical aspects of this circuit are made up of the moment of inertia (Jm) and the viscous friction coefficient (Bm). Other integral parameters include the torque constant (Kt) and the back emf constant (Kv). The corresponding circuit for a DC motor with a permanent magnet is shown in Figure 1.

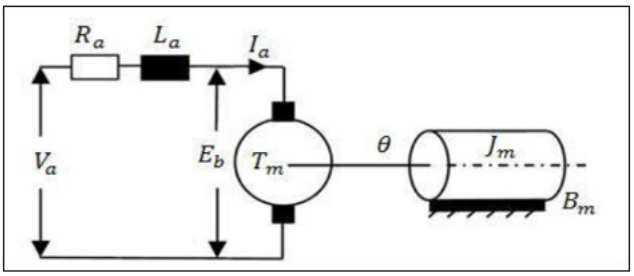

Figure 1. Equivalent circuit of PMDC (Adel et al., 2018)

The Permanent magnet DC motor block diagram is shown in Figure 2.

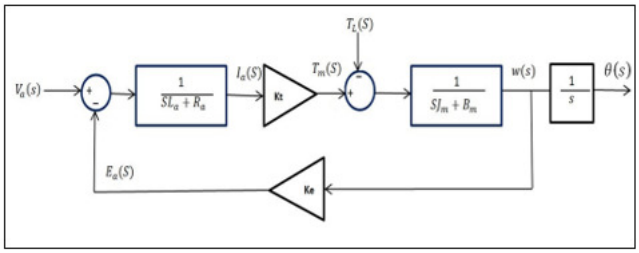

Figure 2. Block diagram of PMDC (Adel et al., 2018)

The mathematical representation for the function shown in Figure 2 can be encapsulated in the Equations (1,2,3 and 4) provided below, as per references (Adel et al. 2018).

$$v_a(t) = e_a(t) + R_a i_a(t) + L_a \frac{d}{dt} ia(t) \quad (1)$$

$$e_a(t) = k_e \omega_m(t) \quad (2)$$

$$T_m(t) - T_L = J_m \frac{a}{dt} \omega_m(t) + B_m \omega_m(t) \quad (3)$$

$$T_m(t) = K_t i_a(t) \quad (4)$$

By using Laplace transformation for the Equations (1-4), Equations (5-8) will be:

$$V_a(s) = E_a(s) + R_a I_a(s) + S L_a I_a(s) \quad (5)$$

$$E_a(s) = k_e \ \omega_m(s) \quad (6)$$

$$T_m(s) - T_L = S J_m \omega_m(s) + B_m \omega_m(s) \quad (7)$$

$$T_m(s) = K_t I_a(s) \quad (8)$$

Equations (9) and (10) define the global transfer functions for speed and position control of the PMDC motor.

$$\frac{\omega_m(s)}{V_a(s)} = \frac{k_e}{JL_aS^2+(JR_a+BL_a)S+BR_a+k_e^{\ 2}} \quad (9)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{k_e}{JL_aS^3+(JR_a+BL_a)S^2+(BR_a+k_e^2)S} \quad (10)$$

Optimization Tuning Methods

This paper employs the classic method (CM), in conjunction with three optimal approaches for determining and extracting PID cascade parameters. These include the genetic algorithm (GA), the dandelion optimizer algorithm (DOA), and the butterfly optimization algorithm (BOA). A comparative analysis will be undertaken among these methods to discern the most beneficial outcomes. Further details about these techniques will be elucidated in the ensuing section.

Classical Method (CM)

The classical method computes the P and PI parameters of the cascade control system by employing block diagram reduction, simplifying each loop independently, and calculating its respective parameters. However, this method isn't feasible for real-time applications due to certain assumptions it makes, such as disregarding the load and negating the back EMF effect (Biswas and Anupam 2013). Initial calculations focus on the parameters of the inner loop or the current loop. Figure 3 illustrates the initial control loop. In a PMDC motor, the torque and current exhibit proportionality, thereby allowing the current to be treated as a control variable.

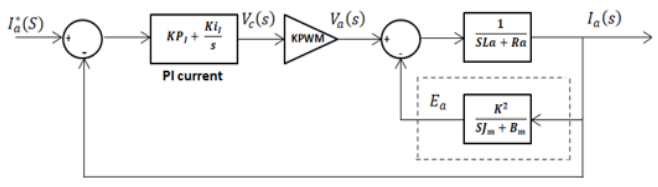

Figure 3. Current loop

For the purpose of clarity, this method simplifies things by making assumptions that lead to important circuit details being ignored. In particular, because of the large magnitude of the motor's moment of inertia (Jm), the effects of the torque load and the counter-electromotive force (Ea) are disregarded. Therefore, these elements have been removed from the current control loop and a simpler form is shown in Figure 4.

The transfer function of The PI current controller as in Equation (11)

$$\frac{V_c(s)}{E(s)} = Kp + \frac{K_i}{s} = \frac{K_i}{s}\left(1 + \frac{s}{{}^{Ki}/_{Kp}}\right) \quad (11)$$

Figure 3 shows a simplification of the current loop, and its open-loop transfer function designated $G_{I\ o_l}$) is provided by Equation (12). This function characterizes the system's operation in the absence of any kind of feedback.

$$G_{I\ ol} = \frac{k_{iI}}{S}\left(1 + \frac{S}{{}^{k_{iI}}/_{k_{pI}}}\right) * KPWM * \left(\frac{{}^{1}/_{R_a}}{1+s\tau_e}\right) \quad (12)$$

Where k_{pI} is the proportional gain of current controller k_{iI} is the integral gain of current controller τ_e is the electrical time constant $\frac{L_a}{R_a}$

By cancellation of the pole in the motor transfer function renders Equation (13) will be:

$$G_{Iol} = \frac{k_{iI} * KPWM}{sR_a} \quad (13)$$

he range or crossover frequency for the current loop, denoted by ci in Equation (14), may be calculated from the open-loop transfer function. At this frequency (ωci), the system's power reduces to half its peak value, denoting the boundary between the passband and the stopband in the frequency response.

$$\omega ci = \frac{KPWM * k_{iI}}{R_a}, \text{ or } \quad k_{iI} = \frac{\omega ci * R_a}{KPWM} \quad (14)$$

The DC-DC converter's switching frequency, fci, may be used in the equation ωci=2πfci to get the bandwidth frequency, fci. In most cases, ci will have a value that is 10 times less than fci. We assume a switching frequency of 2 kHz for the purposes of this work. We assume the closed current loop operates optimally for the purpose of determining the speed loop parameters. Figure 4 is a graphic representation of this idea, which represents harmony.

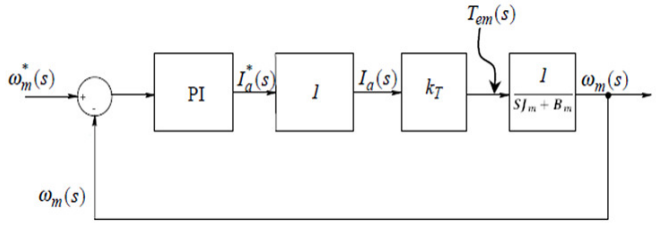

Figure 4. Block diagram of the speed loop

The pole associated with the motor's mechanical component will be cancelled, as described in Equation (15), in a manner analogous to the inner current control loop.

K_{is} represents the integral gain of the controller, whereas K_{ps} specifies the proportional gain. These two variables play crucial roles in how the control system reacts to perturbations or mistakes.

$$\frac{k_{iS}}{k_{pS}} = \frac{1}{\tau_m} \quad (15)$$

The current control loop's bandwidth, written by ci, is often chosen to be an order of magnitude larger than the speed control loop's bandwidth, denoted by cs. To illustrate this connection, consider Equation (16). This setup guarantees a consistent and proportional response from the system's dual control loops.

$$\omega cs = \frac{k_{is} \cdot k_t}{B_m} \quad (16)$$

It is assumed that the velocity loop is perfect, and is therefore represented as unity, in order to calculate the kpP, parameter of the position loop. Equation (17) represents the open-loop transfer function for position control.

$$G_{Pol}(s) = \frac{k_{pP}}{s} \quad (17)$$

Using Equation (18), we may determine the value of the position parameter, kpP, by choosing a bandwidth frequency, denoted by ps, that is ten times lower than the bandwidth frequency, ωps, of the speed control loop. This choice guarantees a consistent hierarchy of control loop frequencies, which boosts the robustness and efficiency of the overall system.

$$\omega pS = k_{pP} \quad (18)$$

Genetic Algorithm Optimization (GA)

The current generation produces the next by way of the evolutionary process. It is anticipated that this next generation would provide a superior outcome. This process is iterated over many generations to get the best possible outcome for the system.

Table 1 represents for GA parameters.

Table 1. GA parameters

GA Parameters	Value
Generation	100
Population size	20
No. dimension	5
Crossover	0.6
LU, UB	0.300

Butterfly Optimization Algorithm (BOA)

The BOA is a metaheuristic algorithm that takes its inspiration from the behavior of butterflies, particularly their use of natural cues for food searching and mating to produce offspring. Chemical receptor nerve cells on the butterfly's body disperse the scent of nectar, which allows other butterflies to detect its presence. The intensity of this fragrance is directly proportional to the butterfly's fitness. The foundational principles of sensing and odor processing can be grouped into three main categories: sensory modality (c), stimulus level (I), and power exponent (a). The natural behaviors exhibited by butterflies inform the formulation of the fragrance function and the variations in scent intensity. The fragrance function can be represented mathematically as a function of physical density, as shown in Equation (19). This equation encapsulates the underlying principles of the algorithm, leveraging the natural behaviors of butterflies to guide the search for optimal solutions..

$$f = cI^a \quad (19)$$

The value of f in the aforementioned equation indicates the intensity with which other butterflies detect the aroma. Perceived scent intensity depends on a number of factors, including the previously discussed metrics *c*, *I*, and *a*.

The butterfly moves in a global search direction when it detects another butterfly's odor, represented by Equation (20). In contrast, when it cannot detect another butterfly's odor, it moves in a random direction, represented by Equation (21).

$$X_i^{t+1} = X_i^t + \left(r^2 * g^* - X_i^t \right) * f_i \quad (20)$$

$$X_i^{t+1} = X_i^t + \left(r^2 * X_j^t - X_k^t \right) * f_i \quad (21)$$

Where X_i^{t+1} represents to new solution vector, X_i^t represents to solution vector, r represents to rondom number (n) between [0, 1], g represents to the current best solution, t is the iteration number, i is the butterfly and X_j^t & X_k^t are *jth* and *kth* butterflies from the solution space and *fi* the perceived magnitude of the frequence.

Butterflies may either look close to home or far and wide for food and companionship. A butterfly may choose between the two different kinds of search methods. It may choose to either stick with the butterfly that is doing the best overall search, or it can wander aimlessly. The'switch probability, represented by the symbol, is the method by which the butterfly may toggle between local and global searches. The BOA parameters are shown in Table 2.

Table 2. BOA parameters

Parameters	Value
Max iteration	5
No. search agents	20
No. dimension	5
Switch probability ρ	0.8
Power exponent a Sensory modality LB, UB	0.1 0.01 0.300

The butterfly optimization technique is able to be depicted in the form of an approximate flowchart, which may be found in Figure 8.

Dandelion Optimizer Algorithm (DOA)

This segment introduces the mathematical formulas utilized in the Dandelion Optimization Algorithm (DOA). It commences with an explanation of two distinct types of conditions and their corresponding mathematical expressions. After this, it examines mathematical simulations for the last phases of the flight, including descent and landing. Equation (22) shows the population array.

$$population = \begin{bmatrix} x_1^1 & \cdots & x_1^{dim} \\ \vdots & \ddots & \vdots \\ x_{pop}^1 & \cdots & x_{pop}^{dim} \end{bmatrix} \quad (22)$$

Here '*pop*' stands for population size, whereas 'Dim' refers to the dimension of the variable in question. The problem's upper limit (UB) and lower bound (LB) are used as constraints to construct a set of randomly generated candidate solutions. According to Equation (23), the ith person's representation (Xi) is generated at random.

$$Xi = rand \times (UB - LB) + LB \quad (23)$$

In the Equation, "i" is an integer that ranges from 1 to "pop", and "rand" represents a randomly generated number between 0 and 1. The lower bound (LB) and upper bound (UB) values are expressed as follows in Equation (24).

$$LB = [lb_1, \ldots\ldots\ldots\ldots, lb_{Dim} \quad (24)$$

$$UB = [ub_1, \ldots\ldots\ldots\ldots, ub_{Dim} \quad (25)$$

The initial elite is determined by DOA's ideal fitness value during the initiation phase. This particular person is said to be the best environment for the dandelion seed to grow. Equations (26) and (27) reveal the mathematical equation for the initial elite *Xelite* if the smallest value is used:

$$f_{best} = min(X_i) \quad (26)$$

$$X_{elite} = X(find(f_{best} == (X_i))) \quad (27)$$

The expression "*find*()" refers to two indices that have the same value. Table 3 presents the parameters of the Dandelion Optimization Algorithm (DOA) used.

Table 3. DOA parameters

Parameters	Value
Max iteration	10
No. search agents	30
No. dimension	5
Switch probability ρ	0.7
Power exponent a Sensory modality LB, UB	0.1 0.01 0.500

Objective Functions

In optimal control theory and the construction of estimators employing linear state variable feedback, the notion of an objective function is fundamental. The goal of the system in such situations is to maximize some measure of performance under strict limits. Functions that rely on error and time are often used as performance indices. The greatest potential performance of the system is ensured by this optimization procedure. This is crucial for improving control systems' efficacy and efficiency.

Integral of absolute error (IAE), integral of squared error (ISE), integral time absolute error (ITAE), and integral time squared error (ITSE) and etc.

In this study the integral time absolute error (ITAE) is used. The ITAE criterion generally produces a smaller overshoot and oscillation than ISE and IAE criteria. In addition, it is the most sensitive of the three, and sometimes too sensitive-slight parameter variation degrades system performance.

ITAE can be represented by the following equation:

$$\text{ITAE} = \int_0^{T_s} t\ |e_\theta(t)|dt + \int_0^{T_s} t\ |e_w(t)|dt + \int_0^{T_s} t\ |e_{Ia}(t \quad (28)$$

Simulation Results and Discussions

The purpose of this research is to identify the best values for the controller's parameters in a cascade P-PI setup, so that a single-axis PMDC motor may be controlled with precision in terms of speed and position. We achieve this by comparing the results of four different tuning strategies: GA, CM, BOA, and the DOA. The findings will provide light on which of these approaches yields the most precise tuning for the controller, and hence better motor control. According to Figures 5, 6, 7 and 8 analysis of the Classical Method (CM) revealed that it did not yield satisfactory results. The system became unstable when parameters extracted from this method were implemented. When the Genetic Algorithm (GA) was applied, a noticeable deviation of approximately 27° from the reference position was observed. Furthermore, the speed appeared irregular, and the motor rotated in the opposite direction before stopping at the desired position. The application of the Butterfly Optimization Algorithm (BOA) resulted in an 18° deviation from the reference position. This also led to irregular speed and the motor reversing its rotation before settling at the desired positionIn contrast, the Dandelion Optimization Algorithm (DOA) showed no deviation from the reference position or speed overshoot.

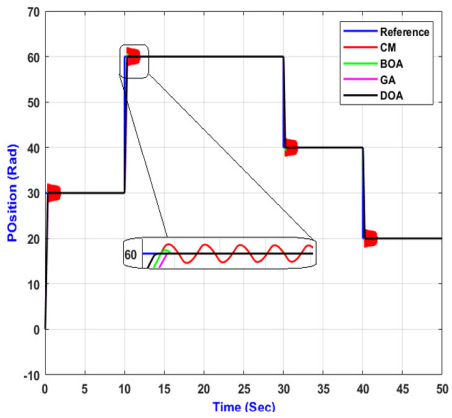

Figure 5. Position based on CM, BOA, GA, and DOA strategies at 17 Nm as load

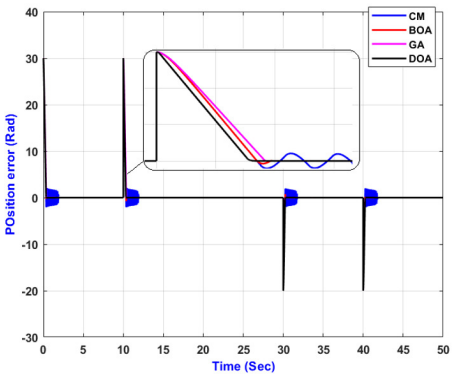

Figure 6. Position error based on CM, BOA, GA, and DOA strategies at 17 Nm as load

The system was rigorously tested under all conditions, including load scenarios, with single or multiple applied reference positions.

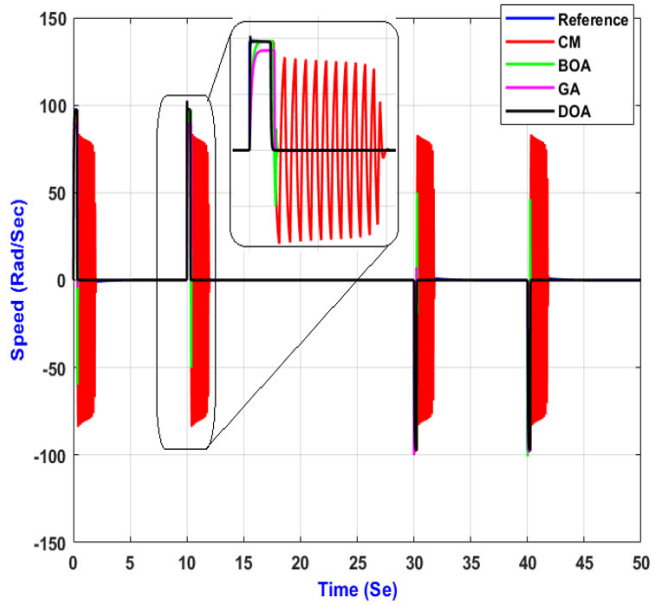

Figure 7. Speed based on CM, BOA, GA, and DOA strategies at 17 Nm as load

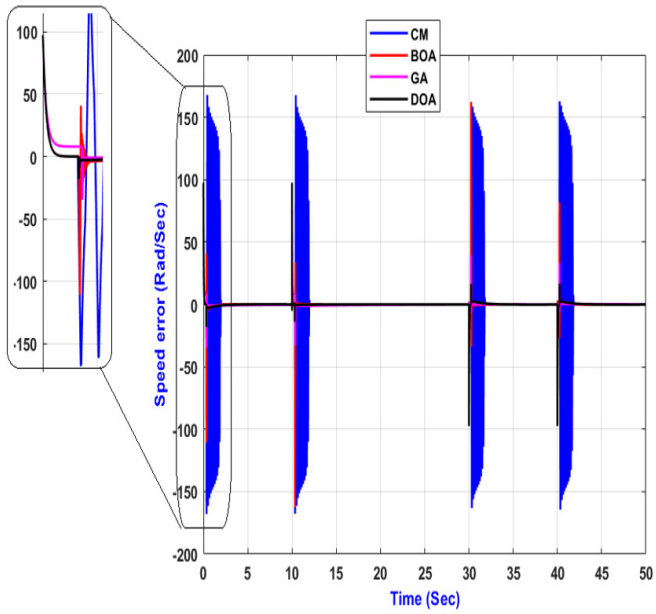

Figure 8. Speed error result based on CM, BOA, GA, and DOA strategies at 17 Nm as load

The system was rigorously tested under all conditions, including load scenarios, with single or multiple applied reference positions. The DOA algorithm consistently demonstrated accurate trajectory tracking to reach the desired position at a steady speed, Notably, the voltage and current values remained within the safe limits, preventing potential damage to the motor. Table 4 provides a detailed comparison of the performance criteria values for the position simulation results under full load conditions for the three optimal algorithms (BOA, DOA, and GA).

Table 4: Performance criteria

Performance criteria	GA	DOA	BOA
Rising time (sec)	0.35	0.32	0.6
Settling time (sec)	0.5	0.4	0.4
Overshoot %	8.1128	4.796	2

The Table 5 explains the values of P-PI parameters that we obtaineded from the application of different optimization methods.

Table 5: Cascade P-PI parameters

Method	KP position	KP speed	KI speed	KP current	KI current
CM	62.8319	2.0944	0.9425	0.3817	21.153
GA	186.4578	133.669	11.887	55.6998	2.365
BOA	24.2032	26.985	38.9883	8.79824	46.0113
DOA	44.0937	63.3902	19.4847	70.639	3.21979

CONCLUSION

The results of this study significantly contribute to the field of motor control systems by introducing a robust cascade P-PI controller for effective speed and position control of single-axis PMDC motors. The controller, primarily based on simulation design, was developed and tested using both classical methods (CM) and three advanced optimization techniques: Genetic Algorithm (GA), Dandelion Optimization Algorithm (DOA), and Butterfly Optimization Algorithm (BOA). The performance evaluation conducted under different operating conditions, including load and no-load scenarios, further corroborated the superiority of the DOA technique. In particular, the DOA excelled by delivering no deviation from the reference position or speed overshoot and keeping voltage and current values within permissible limits. This enhanced safety feature signifies the potential of the proposed controller to prevent possible motor damage. The comparative analysis performed in this study emphasized the drawbacks of the CM and the inconsistency of the GA and BOA. Meanwhile, it spotlighted the effectiveness and resilience of the DOA in tuning the controller's parameters, consistently achieving accurate trajectory tracking and the desired position at a steady speed. So, the proposed cascade P-PI controller, combined with the Dandelion Optimization Algorithm, offers a robust and efficient solution for controlling the speed and position of single-axis PMDC motors. It opens new avenues for further research in the area of motor control systems, particularly those that focus on enhancing accuracy, resilience, and safety.

REFERENCES

1. Latif, A. (2020). Optimum Synthesis of a BOA optimized novel dual-stage PI-(1+ ID) controller for frequency response of a microgrid. Energies, 13(13), 3446.
2. Ali, W. H., Sadiku, M. N. O., & Abood, S. (2019). Fundamentals of Electric Machines: A Primer with MATLAB: A Primer with MATLAB. CRC Press.
3. Bigelow, T. A. (2020). Electric Circuits, Systems, and Motors. Springer.
4. Cankurtaran, M. F., & Kocamis, A. E. (2019). Sensorless Speed Control of PMDC Motor with Cascade PI Controller. International Symposium ELMAR. IEEE.
5. Chau, K. T., & Wang, Z. (2011). Chaos in electric drive systems: analysis, control and application. John Wiley & Sons.
6. Cuong, N. D., & Puta, H. (2013). Design of MRAS based control systems for load sharing of two DC motors with a common stiff shaft. 2013 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE.
7. Dorf, R. C., & Bishop, R. H. (2011). Modern control systems. Pearson.
8. Franchi, C. M. (2019). Electrical Machine Drives: Fundamental Basics and Practice. CRC Press.
9. Ishizaki, K., Sencer, B., & Shamoto, E. (2013). Cross Coupling Controller for Accurate Motion Synchronization of Dual Servo Systems.
10. Krause, P. C., Wasynczuk, O., Sudhoff, S. D., & Pekarek, S. (2002). Analysis of electric machinery and drive systems (3rd ed., Vol. 2). Wiley Online Library.
11. Mehta, V. K., & Mehta, R. (2008). Principles of electrical machines. S. Chand Publishing.
12. Mohan, N. (2003). Electric drives.

13. Momoh, J. A. (2018). Energy Processing and Smart Grid. John Wiley & Sons.
14. Namazov, M., & Basturk, O. (2010). DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods.
15. Panlong, Z., & Wang, Z. (2019). Improvements of direct current motor control and motion trajectory algorithm development for automated guided vehicle. *Advances in Mechanical Engineering*, 11(2), 1687814018824937.
16. Pisano, A. (2008). Cascade control of PM DC drives via second-order sliding-mode technique. IEEE *Transactions on Industrial Electronics*, 55(11), 3846-3854.
17. Raja, G. L., & Ali, A. (2017). Series cascade control: An outline survey. 2017 Indian Control Conference (ICC). IEEE.
18. Romero, L., & Antonio, C. (2006). Control of position/velocity in a mobile robot using dc brushless motors. Electronics, Robotics and Automotive Mechanics Conference. IEEE.
19. Son, Y. I. (2014). Robust cascade control of electric motor drives using dual reduced-order PI observer. IEEE *Transactions on Industrial Electronics*, 62(6), 3672-3682.
20. Taha, N., & Gücin. (2015). Tuning cascade PI (D) controllers in PMDC motor drives: A performance comparison for different types of tuning methods. 2015 9th International Conference on Electrical and Electronics Engineering (ELECO), IEEE.
21. Talavaru, U., & Naik, R. N. (2014). Microcontroller based closed loop speed and position control of DC motor.
22. Tang, J. (2001). PID controller using the TMS320C31 DSK with online parameter adjustment for real-time DC motor speed and position control. ISIE. IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570), 2, IEEE.
23. Taut, E., & Marius, A. (2018). Model-in-the-Loop for Determining the Speed and Position of a DC Motor. 41st International Spring Seminar on Electronics Technology (ISSE). IEEE.
24. Wang, L. (2020). PID control system design and automatic tuning using MATLAB/Simulink. John Wiley & Sons.

Khalaf Abdullah Khalaf KHALAF

Khalaf Abdullah Khalaf Khalaf is a dedicated. Master's student specializing in power electronics, optimization and control systems. Khalaf's academic journey started with a Bachelor's degree in Electrical Engineering. His innovative work revolves around creating advanced algorithms and control mechanisms to improve energy utilization and system performance. He is driven by the goal to contribute substantially to the field and enhance the efficiency and reliability of power electronic systems.